

Министерство высшего образования и науки РФ
ФГБОУ ВО ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Кафедра «Информационная безопасность систем и технологий»

ОТЧЁТ

о лабораторной работе №3

Использование стандартной библиотеки C++ при работе со строками

Дисциплина: Языки программирования

Группа: 18ПИ1

Выполнил: Новиков Д.О

Количество баллов:

Дата сдачи:

Проверил: к.т.н., доцент Лупанов М.Ю.

Пенза, 2019

1 Цель работы

1.1 Освоить работу со строками с помощью стандартной библиотеки C++.

2 Задания к практической работе

2.1 Разработать программу, запрашивающую у пользователя целое число в диапазоне от 1 до 100, а затем выводящую фразу «В караване было X верблюдов», где вместо X указать введенное число. Программа должна использовать корректные падежные окончания. Программа должна быть реализована с использованием строк Си++.

2.2 Разработать программу, заменяющую в тексте из Приложения Д слово `merhi` на слово `yandex`. Программа должна быть реализована с использованием строк Си++.

2.3 Разработать программу, подсчитывающую в тексте из Приложения Д количество букв, цифр и прочих знаков. Пробелы и символы новой строки не учитывать. Программа должна быть реализована с использованием строк Си++.

2.4 Реализовать функцию, выполняющую зашифрование методом Цезаря. Функция должна иметь следующий прототип `std::wstring encrypt(const std::wstring& text, const int key)`, где `text` — открытый текст на русском языке, `key` — величина сдвига (ключ). Функция должна перед зашифрованием удалить из текста все символы, кроме русских букв, и привести их к верхнему регистру. Функция должна вернуть строку с зашифрованным текстом.

2.5 Реализовать функцию, выполняющую расшифрование методом Цезаря. Функция должна иметь следующий прототип `std::wstring decrypt(const std::wstring& text, const int key)`, где `text` — зашифрованный текст на русском языке, `key` — величина сдвига (ключ). Функция должна вернуть строку с расшифрованным текстом в случае успеха или строку со словом ОШИБКА при невозможности расшифровать текст (в зашифрованном тексте есть недопустимые символы) .

3 Результат выполнения работы

3.1 Была разработана программа, запрашивающая у пользователя целое число от 0 до 100. После ввода числа пользователем числа, оно проверяется на соответствии условию. Если число не проходит проверку, то программа завершается. Если проверка пройдена, то происходят проверки числа для выбора верного падежного окончания слов. Если остаток от деления введённого числа на 10 равен 1 и это число больше или равно 21 или равно 1, то стирается окончание «ов» в слове «верблюдов», число записывается в строку, удаляется символ «о» в слове «было». Если остаток от деления числа на 10 больше 1 и меньше 5 и число больше или равно 21, окончание «ов» в слове «верблюдов», в строку записывается введённое число, после чего к слову «верблюд» прибавляется «а». Если остаток от деления числа на 10 равен 0 или больше или равен 5, то в строку записывается число. Блок-схема, описывающая алгоритм работы программы, изображена на рисунке 1. Код программы на рисунке 2 и результат работы на рисунке 3.

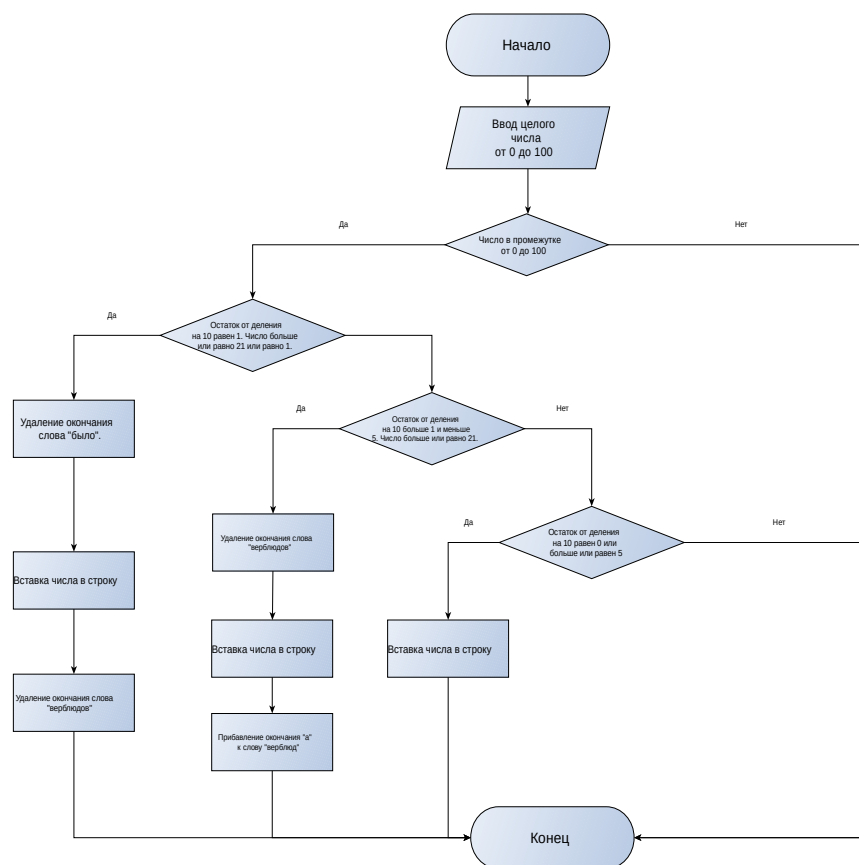


Рисунок 1-Блок-схема

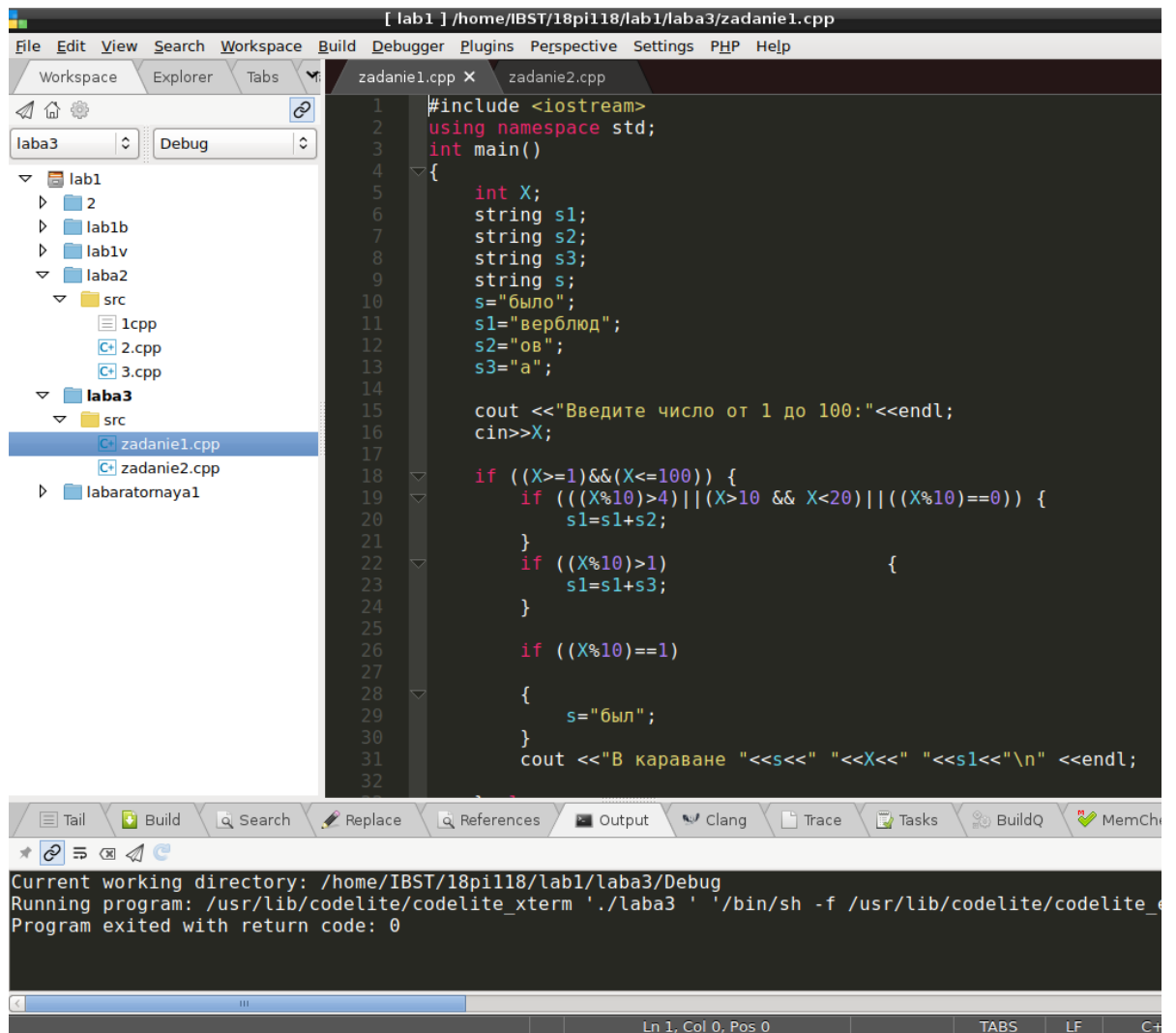
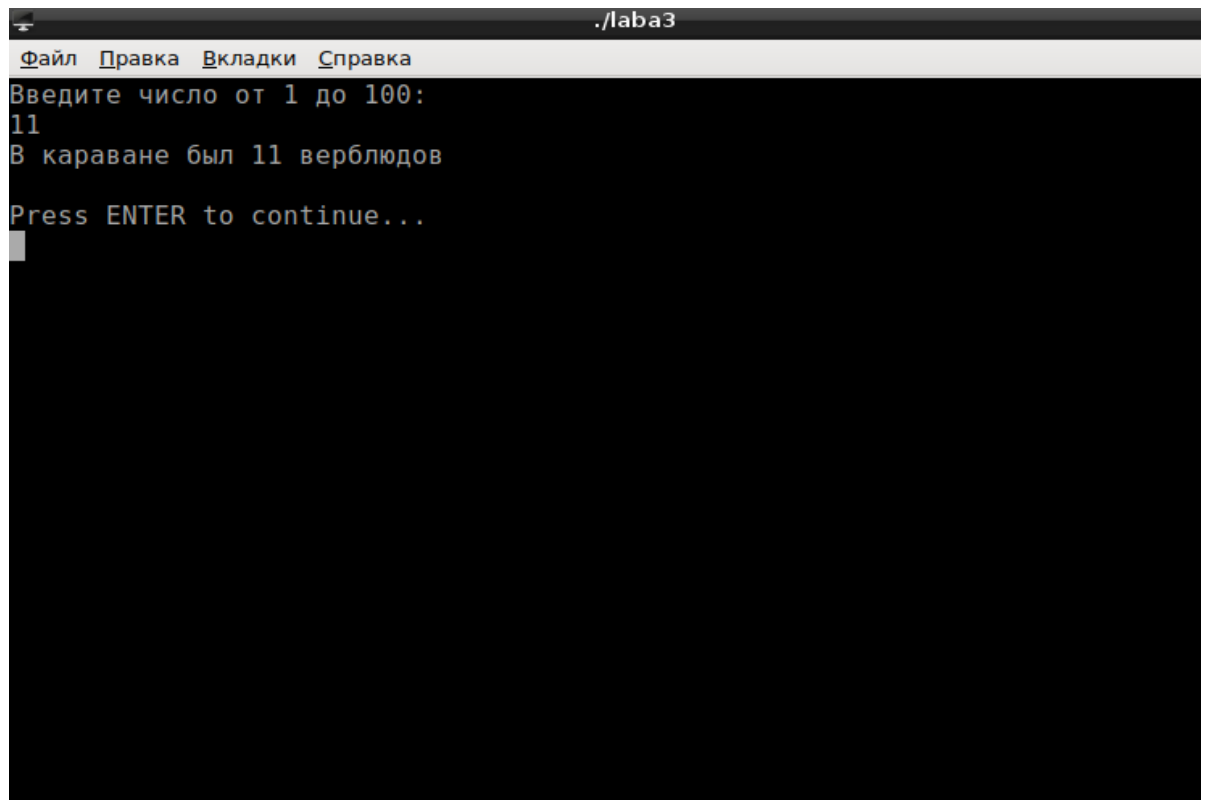


Рисунок 2-Код программы



```
./laba3
Файл  Правка  Вкладки  Справка
Введите число от 1 до 100:
11
В караване был 11 верблюдов
Press ENTER to continue...
```

Рисунок 3-Результат работы

3.2 Была разработана программа для замены слова «merhi» на слово «yandex» в тексте из приложения Д. Программа ищет позицию первого символа слова «merhi» в строке, если позиция найдена, то слово «merhi» заменяется словом «yandex». Поиск продолжается до тех пор, пока программе не удастся обнаружить первый символ слова «merhi». Блок-схема описывающая алгоритм работы программы, представлена на рисунке 4.

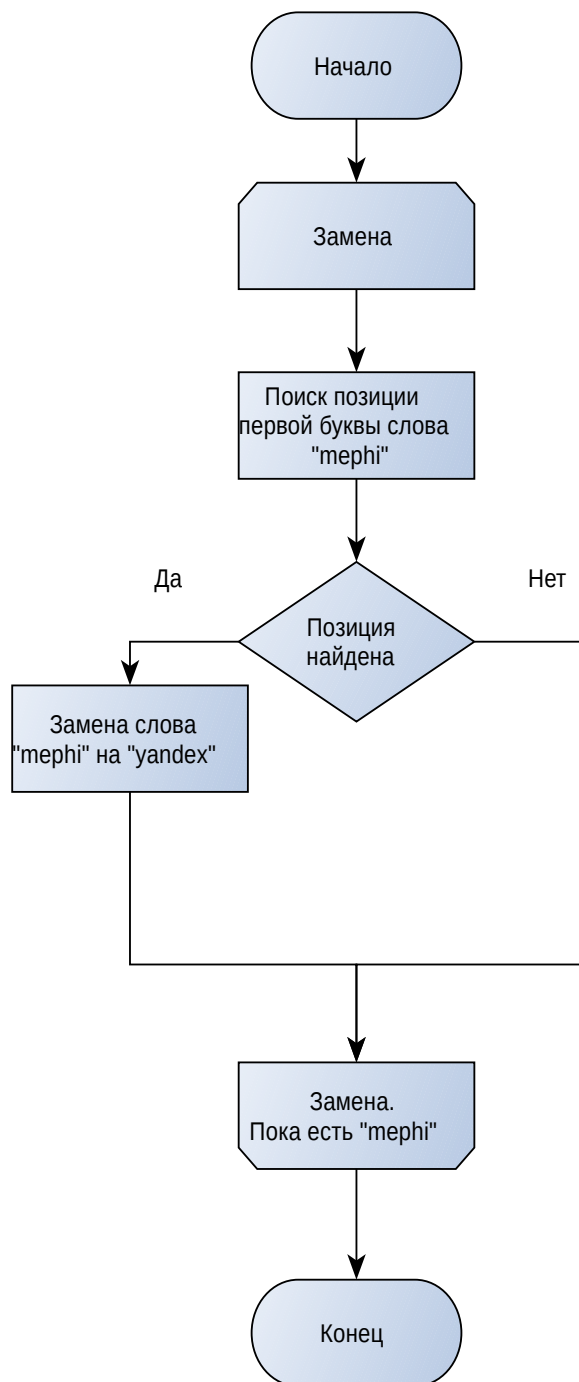


Рисунок 4-Блок-схема 2.

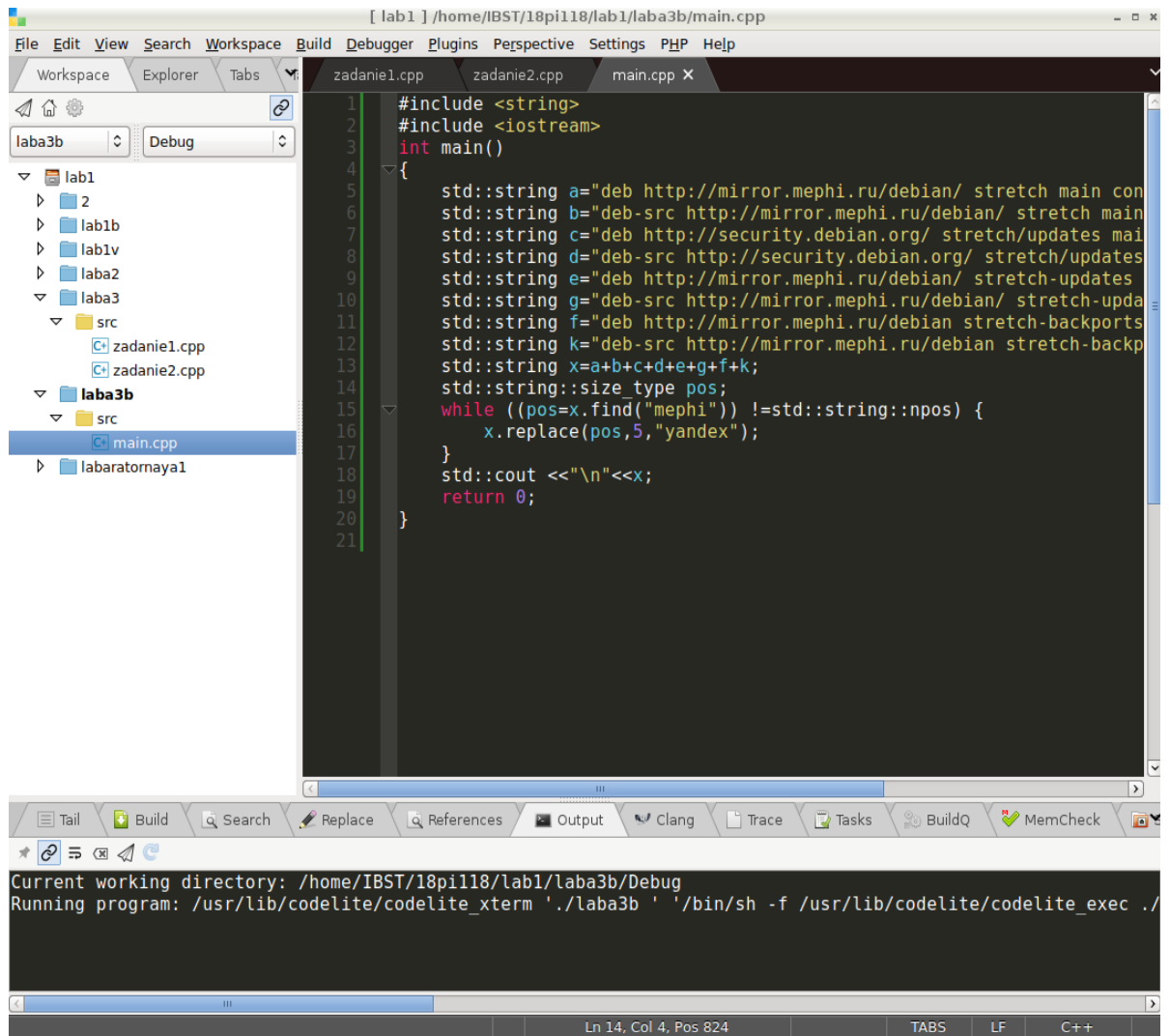


Рисунок 5-Код программы

```
./lab3b
Файл  Правка  Вкладки  Справка
deb http://mirror.yandex.ru/debian/ stretch main contrib non-free
deb-src http://mirror.yandex.ru/debian/ stretch main contrib non-free
deb http://security.debian.org/ stretch/updates main contrib non-free
deb-src http://security.debian.org/ stretch/updates main contrib non-free
deb http://mirror.yandex.ru/debian/ stretch-updates main contrib non-free
deb-src http://mirror.yandex.ru/debian/ stretch-updates main contrib non-f
deb http://mirror.yandex.ru/debian stretch-backports main contrib non-free
deb-src http://mirror.yandex.ru/debian stretch-backports main contrib non-
Press ENTER to continue...
█
```

Рисунок 6-Результат выполнения программы для задания 2

3.3 Была разработана программа, которая считает количество символов в строке из приложения Д, исключая символ пробел и символ новой строки. Программа считает количество всех символов строки, после запускается цикл, который работает, пока счётчик меньше количества символов. В цикле каждый символ проверяется на равенство пробелу или символу новой строки, если он не равен, то искомое количество символов увеличивается на один, если равен, то проверяется следующий символ. Блок-схема, описывающая алгоритм работы программы, представлена на рисунке 7.

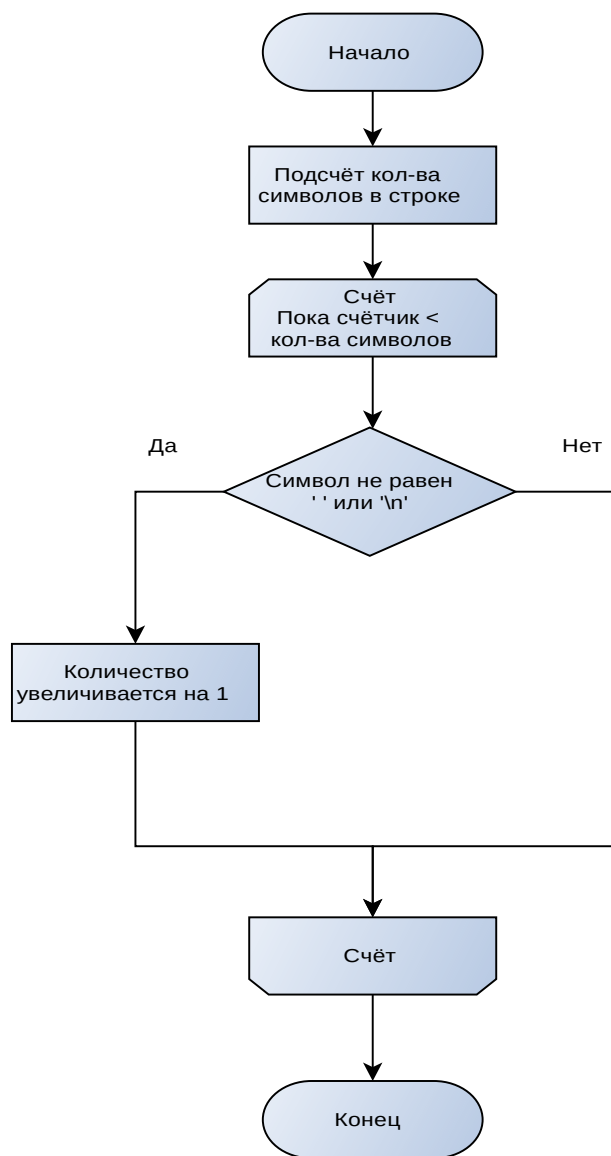


Рисунок 7-Блок-схема 3

Код программы:

```

#include <iostream>
#include <string>
using namespace std;
int main(int argc, char **argv)
{
    int z;
    string str="deb http://mirror.mephi.ru/debian/
stretch          main          contrib          non-free\ndeb-src
http://mirror.mephi.ru/debian/ stretch main contrib non-
free\ndeb  http://security.debian.org/    stretch/updates

```

```

main          contrib          non-free\ndeb-src
http://security.debian.org/ stretch/updates main contrib
non-free\ndeb http://mirror.mephi.ru/debian/ stretch-
updates      main          contrib          non-free\ndeb-src
http://mirror.mephi.ru/debian/ stretch-updates main
contrib      non-free\ndeb http://mirror.mephi.ru/debian
stretch-backports main contrib non-free\ndeb-src
http://mirror.mephi.ru/debian stretch-backports main
contrib non-free\n";
    z=str.size();
    cout<<"Количество символов: ";
    cout<<z<<endl;
    int k=0;
    for(int i=0; i<z; i++) {

        if(str[i]!=' ' and str[i]!='\n'){
            k++;
        }
    }
    cout<<"Количество символов без пробелов и
символов конца строки: ";
    cout<<k<<endl;

    return 0;
}

```

3.4 Была реализована функция, выполняющая шифрование текста шифром Цезаря. Функция получает адрес строки и ключ. Определяется размер строки. Начинается цикл, который работает, пока не переберутся все символы строки. В цикле каждый символ проверяется, если это символ русского алфавита, он приводится к верхнему регистру и передаётся функции для шифрования символа. В функции для шифрования символа запускается цикл, который работает, пока не сравнит полученный символ со всеми символами русского алфавита. Если символ совпадает, то в случае если его позиция в русском алфавите сдвинутая на ключ вправо меньше 33, то в функция вернёт новый символ, стоящий в русском алфавите в позиции полученного символа, сдвинутой вправо на ключ. Если сдвинутая позиция больше или равна 33, то функция возвращает новый символ, который в русском алфавите стоит в

позиции старого символа, сдвинутой на ключ, минус 33. В основной функции шифрования строки, новая строка складывается из символов полученных из функции шифрования символов. Блок - схемы алгоритмов работы функций представлены на рисунках 8 и 9. Код основной функции шифрования строки:

```
wstring encrypt(const wstring & text, const int key)
{
    locale loc("ru_RU.UTF-8");
    locale::global(loc);
    wstring_convert<codecvt_utf8<wchar_t>, wchar_t>
codec;
    wchar_t x=0;
    wstring ntext;
    int z=text.size();
    for(int i=0; i<z; i++) {
        if((text[i]<L'A' or text[i]>L'Я') and
text[i]!=L'Ё' )
            if ((text[i]<L'a' or text[i]>L'я') and
text[i]!=L'ё') {
                continue;
            }
        x=text[i];
        x=toupper(x,loc);
        x=encryptSimvola(x,key);
        ntext=ntext + x;
    }
```

Код функции шифрования символа:

```
wchar_t encryptSimvola(wchar_t x, const int key)
{
    locale loc("ru_RU.UTF-8");
    locale::global(loc);
    wstring_convert<codecvt_utf8<wchar_t>, wchar_t>
codec;
    wstring m= L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
    for(int k=0; k<33; k++) {
        if(x==m[k]) {
            if(k+key<33) {
                x=m[k+key];
                return x;
            } else {
                x=m[k+key-33];
            }
        }
```

```

        return x;
    }
}
return x;
}

```

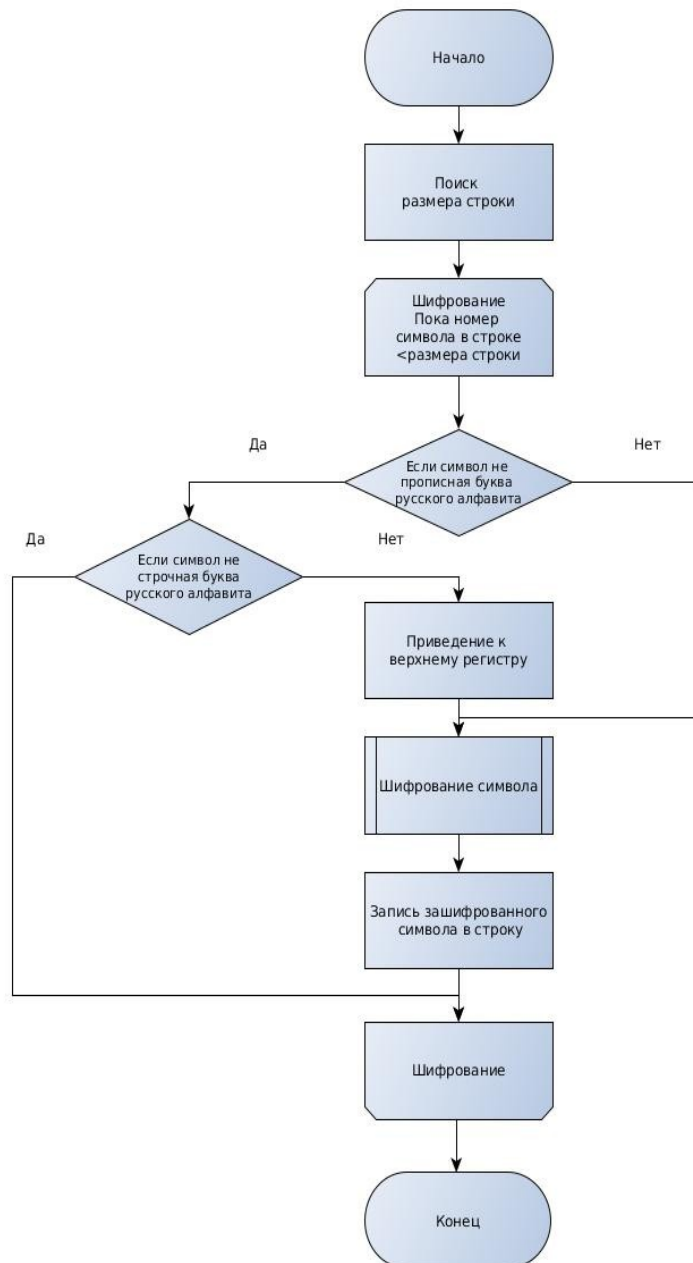


Рисунок 8-Основная функция шифрования строки.

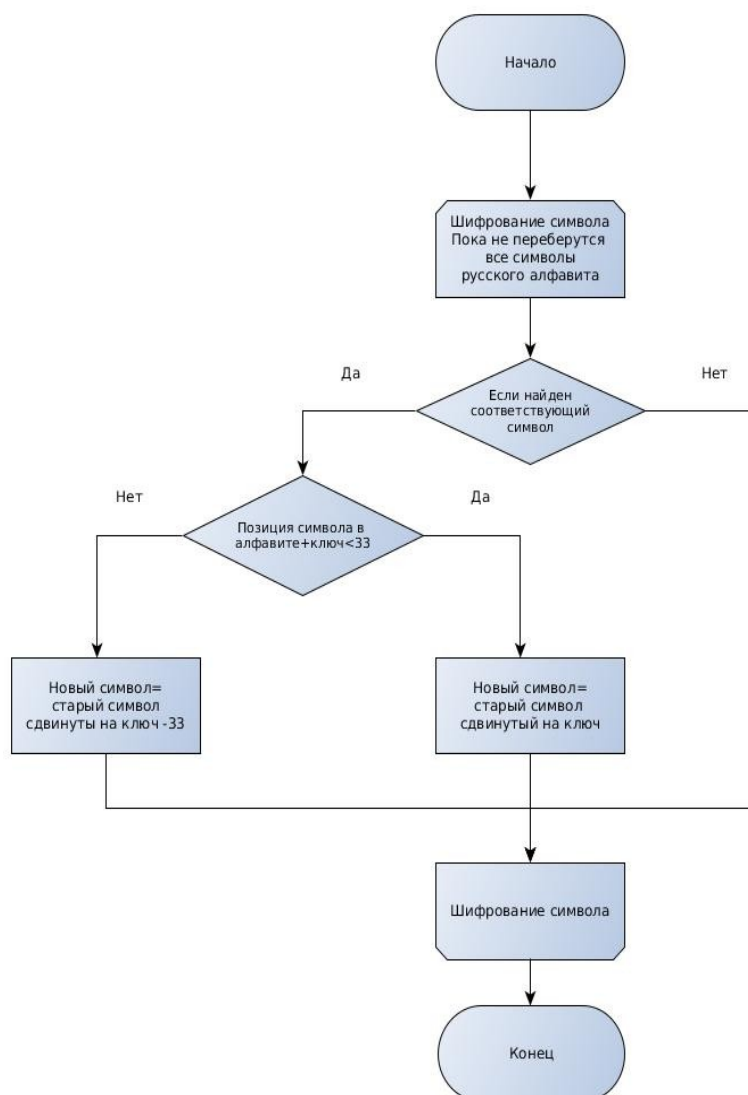


Рисунок 9-Шифрование символа.

3.5 Была реализована функция, выполняющая расшифровку текста, зашифрованного шифром Цезаря. Функция получает адрес строки и ключ. Определяется размер строки. Начинается цикл, который работает, пока не переберутся все символы строки. В цикле каждый символ проверяется, если это символ русского алфавита, он приводится к верхнему регистру и передаётся функции для расшифровки символа. Если в строке есть недопустимый символ, то функция возвращает строку «ОШИБКА!». В функции для расшифровки символа запускается цикл, который работает, пока не сравнит полученный символ со всеми символами русского алфавита. Если символ совпадает, то в

случае если его позиция в русском алфавите сдвинутая на ключ влево больше или равна нулю, то в функция вернёт новый символ, стоящий в русском алфавите в позиции полученного символа, сдвинутой влево на ключ. Если сдвинутая позиция меньше 0, то функция возвращает новый символ, который в русском алфавите стоит в позиции старого символа, сдвинутой вправо на 33 минус ключ. В основной функции расшифровки строки, новая строка складывается из символов полученных из функции расшифровки символа. Блок — схемы алгоритмов работы функций представлены на рисунках 10 и 11.

Код основной функции расшифровки строки:

```
wstring decrypt(const wstring & text, const int key)
{
    locale loc("ru_RU.UTF-8");
    locale::global(loc);
    wstring_convert<codecvt_utf8<wchar_t>, wchar_t>
codec;
    wchar_t x=0;
    wstring ntext;
    int z=text.size();
    for(int i=0; i<z; i++) {
        if((text[i]<L'A' or text[i]>L'Я') and
text[i]!=L'Ё' )
            if ((text[i]<L'a' or text[i]>L'я') and
text[i]!=L'ё' ) {
                ntext=L"ОШИБКА!";
                break;
            }
        x=text[i];
        x=toupper(x,loc);
        x=decryptSimvola(x,key);
        ntext=ntext + x;
    }
    return ntext;
}
```

Код функции расшифровки символа:

```
wchar_t decryptSimvola(wchar_t x, const int key)
{
    locale loc("ru_RU.UTF-8");
    locale::global(loc);
```

```

wstring_convert<codecvt_utf8<wchar_t>, wchar_t>
codec;

wstring m= L"АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";
for(int k=0; k<33; k++) {
    if(x==m[k]) {
        if(k-key>=0) {
            x=m[k-key];
            return x;
        } else {
            x=m[33+k-key];
            return x;
        }
    }
}
return x;
}

```

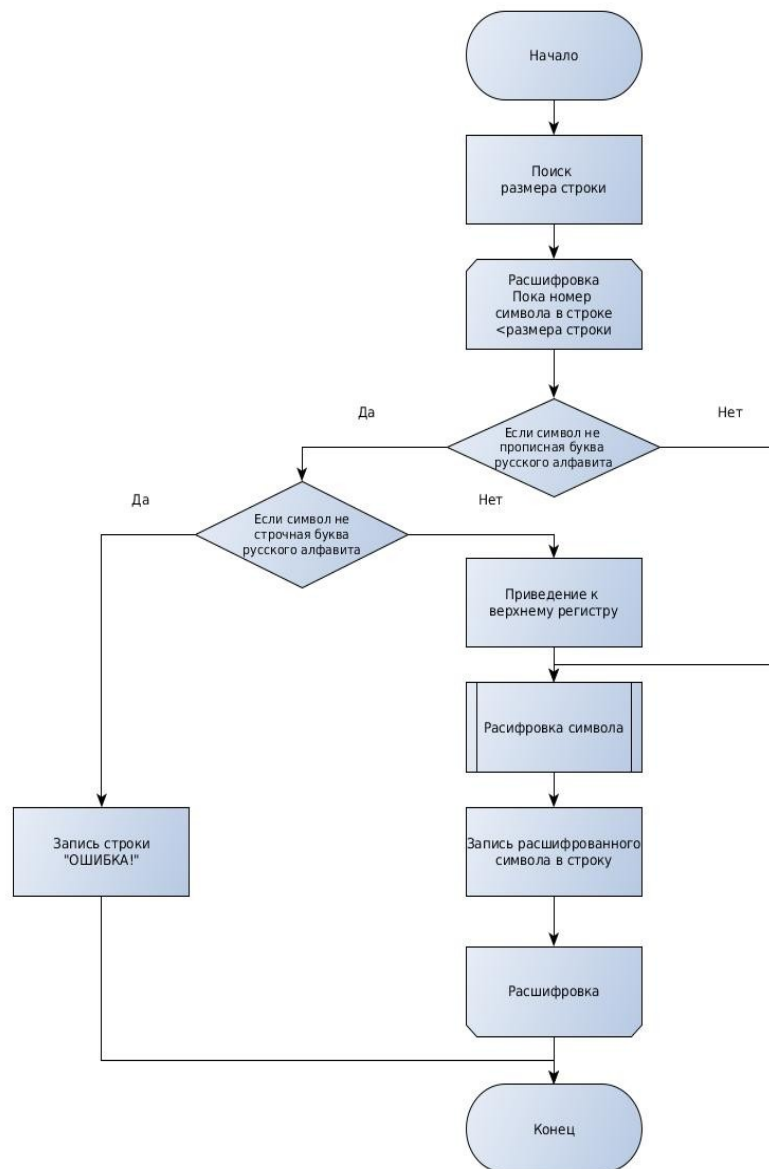


Рисунок 10-Основная функция расшифровки строки

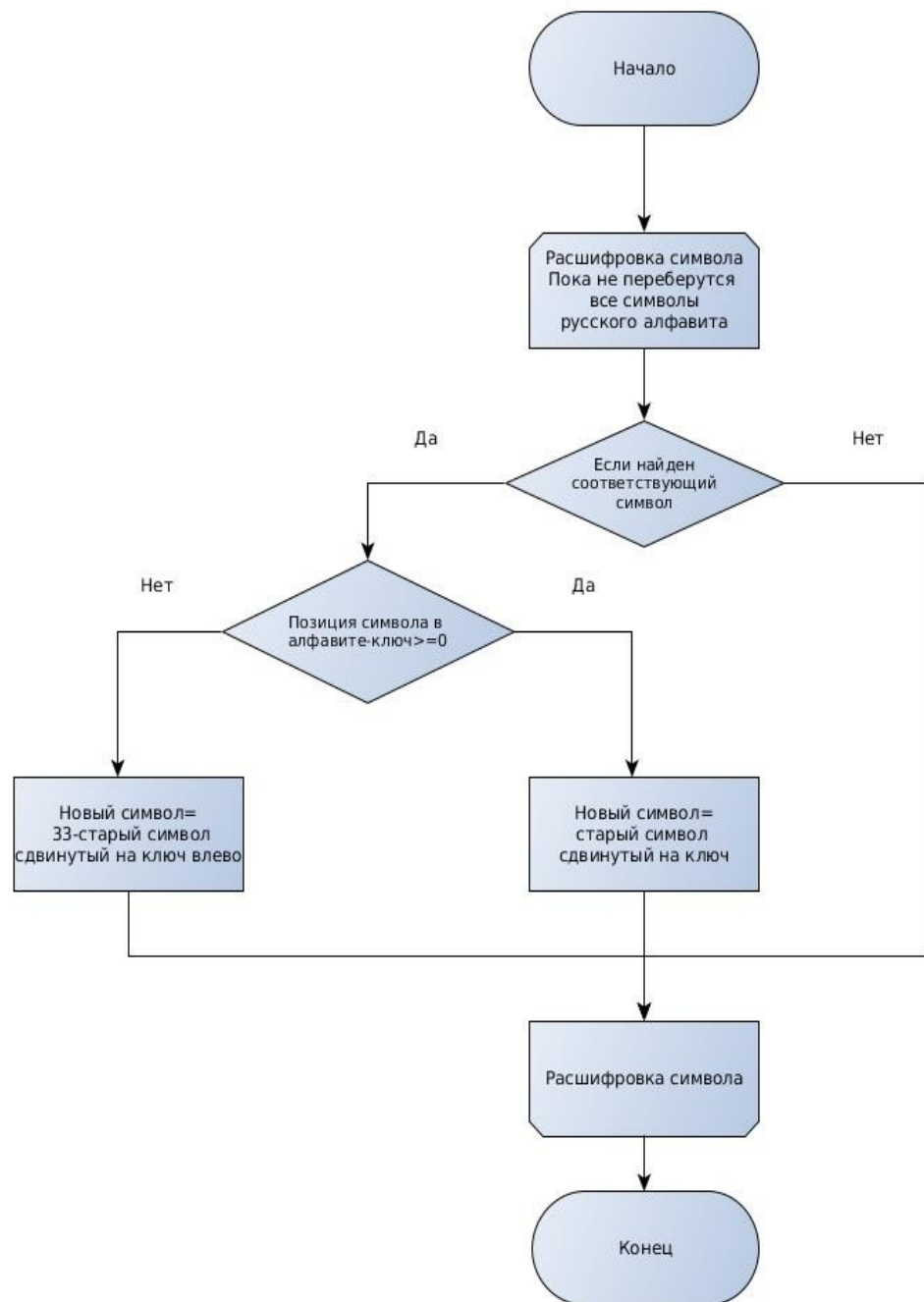


Рисунок 11-Расшифровка символа.

4 Вывод

В процессе выполнения лабораторной работы были изучены строки в языке программирования C++, было освоено использование стандартной библиотеки C++ для работы со строками, были получены практические навыки по использованию различных методов класса string, среди которых методы:

`size()`, `erase()`, `replace()`. Были получены практические навыки по смене кодировок.